

1        **Amendments to the Claims:**

2        This listing of claims is to replace all prior versions and listings of claims in  
3        this application:

4  
5        1.     (Currently Amended) A method comprising:  
6        receiving an original digital good; and  
7        randomly applying various forms of protection to a plurality of segments of  
8        the original digital good to produce a protected digital good,  
9        wherein at least two of the segments overlap one another, wherein  
10       overlapping segments are different from each other but include some identical  
11       data.

12  
13       2.     (Original) A method as recited in claim 1, wherein the randomly  
14       applying comprises pseudo randomly applying the various forms of protection  
15       according to pseudo random techniques.

16  
17       3.     (Original) A method as recited in claim 1, wherein the applying  
18       comprises randomly selecting the forms of protection from a set of available forms  
19       of protection.

20  
21       4.     (Original) A method as recited in claim 1, wherein the applying  
22       comprises applying the various forms of protection to randomly selected portions  
23       of the original digital good.  
24  
25

1           5. (Original) A method as recited in claim 1, wherein the various forms  
2 of protection are selected from a group of protection tools comprising code  
3 integrity verification, acyclic code integrity verification, cyclic code integrity  
4 verification, secret key scattering, obfuscated function execution,  
5 encryption/decryption, probabilistic checking, Boolean check obfuscation, in-  
6 lining, reseeding pseudo random number generators with time varying inputs, anti-  
7 disassembly methods, varying execution paths between runs, anti-debugging  
8 methods, and time/space separation between tamper detection and response.

9  
10           6. (Original) A method as recited in claim 1, wherein the applying  
11 comprises applying a form of protection in which a checksum can be computed on  
12 a set of bytes of the digital good without actually reading the bytes.

13  
14           7. (Original) A computer-readable medium comprising computer-  
15 readable instructions that, when executed by a processor, direct a computer system  
16 to perform the method as recited in claim 1.

17  
18           8. (Currently Amended) A method comprising:  
19 segmenting a digital good into a plurality of segments;  
20 selecting multiple segments from the plurality of segments; and  
21 transforming only the selected segments according to different protection  
22 techniques to produce a protected digital good having a composite of variously  
23 protected segments.

1           9.    (Original) A method as recited in claim 8, wherein at least two of the  
2 segments overlap one another.

3  
4           10.   (Original) A method as recited in claim 8, wherein the selecting  
5 comprises randomly selecting the segments.

6  
7           11.   (Original) A method as recited in claim 8, wherein the transforming  
8 comprises transforming the selected segments according to randomly chosen  
9 protection techniques.

10  
11          12.   (Original) A method as recited in claim 8, wherein the transforming  
12 comprises:

13           augmenting at least one segment using a certain protection technique; and  
14           inserting a checkpoint, which may be used to evaluate a validity of the  
15 augmented segment, within the protected digital good but outside of the  
16 augmented segment being evaluated.

17  
18          13.   (Original) A method as recited in claim 8, further comprising  
19 receiving quantitative parameters indicative of how much the protected digital  
20 good should be altered.

21  
22          14.   (Original) A method as recited in claim 13, wherein the  
23 transforming is performed to satisfy the quantitative parameters.

1           15.   (Original) A method as recited in claim 8, wherein the protection  
2 techniques are selected from a group of protection tools comprising code integrity  
3 verification, acyclic code integrity verification, cyclic code integrity verification,  
4 secret key scattering, obfuscated function execution, encryption/decryption,  
5 probabilistic checking, Boolean check obfuscation, in-lining, reseeding pseudo  
6 random number generators with time varying inputs, anti-disassembly methods,  
7 varying execution paths between runs, anti-debugging methods, and time/space  
8 separation between tamper detection and response.

9  
10           16.   (Original) A method as recited in claim 8, wherein the transforming  
11 comprises applying a protection technique in which a checksum can be computed  
12 on a set of bytes of the digital good without actually reading the bytes.

13  
14           17.   (Original) A computer-readable medium comprising computer-  
15 readable instructions that, when executed by a processor, direct a computer system  
16 to perform the method as recited in claim 8.

17  
18           18.   (Original) A method comprising:  
19           establishing parameters prescribing a desired quantity of protection to be  
20 applied to a software product;  
21           parsing the software product into code sections;  
22           selecting at least one code section;  
23           augmenting the selected code section to add protection qualities; and  
24           repeating the selecting and the augmenting for different code sections until  
25 the desired quantity of protection has been applied.

1  
2 19. (Original) A method as recited in claim 18, wherein the establishing  
3 comprises enabling a user to enter the parameters.

4  
5 20. (Original) A method as recited in claim 18, wherein the augmenting  
6 comprises applying a protection technique selected from a group of protection  
7 techniques comprising code integrity verification, acyclic code integrity  
8 verification, cyclic code integrity verification, secret key scattering, obfuscated  
9 function execution, encryption/decryption, probabilistic checking, Boolean check  
10 obfuscation, in-lining, reseeding pseudo random number generators with time  
11 varying inputs, anti-disassembly methods, varying execution paths between runs,  
12 anti-debugging methods, and time/space separation between tamper detection and  
13 response.

14  
15 21. (Original) A method as recited in claim 18, wherein the augmenting  
16 comprises applying a protection technique in which a checksum can be computed  
17 on a set of bytes of the digital good without actually reading the bytes.

18  
19 22. (Original) A computer-readable medium comprising computer-  
20 readable instructions that, when executed by a processor, direct a computer system  
21 to perform the method as recited in claim 18.

22  
23 23. (Currently Amended) A production system, comprising:  
24 a memory to store an original digital good; and  
25

1 a production server equipped with a set of multiple protection tools that  
2 may be used to augment the original digital good for protection purposes, the  
3 production server being configured to parse the original digital good and apply  
4 protection tools selected from the set of protection tools only to selected portions  
5 of the original digital good in a random manner to produce a protected digital good  
6 having a composite of the protected selected portions.

7  
8 24. (Original) A production system as recited in claim 23, wherein the  
9 protection tools are selected from a group of protection tools comprising code  
10 integrity verification, acyclic code integrity verification, cyclic code integrity  
11 verification, secret key scattering, obfuscated function execution,  
12 encryption/decryption, probabilistic checking, Boolean check obfuscation, in-  
13 lining, reseeding pseudo random number generators with time varying inputs, anti-  
14 disassembly methods, varying execution paths between runs, anti-debugging  
15 methods, and time/space separation between tamper detection and response.

16  
17 25. (Original) A production system as recited in claim 23, wherein the  
18 production server applies a protection tool that enables a checksum to be  
19 computed on a set of bytes of the digital good without actually reading the bytes.

20  
21 26. (Original) A production system as recited in claim 23, wherein the  
22 production server has a pseudo random generator to introduce randomness into the  
23 application of the protection tools to various portions of the original digital good.  
24  
25

1       27.   (Currently Amended) An obfuscation system, comprising:  
2       a parser to parse a digital good into a plurality of segments;  
3       a set of protection tools that may be applied to the segments of the digital  
4   good to augment the segments with protection qualities;  
5       a target segment selector to select at least one segment from the plurality of  
6   segments; and  
7       a tool selector to select at least one protection tool from the set of protection  
8   tools and apply the selected protection tool to the selected segment so that a  
9   protection tool of the set of protection tools is applied only to a selected segment  
10   of the plurality of segments.

11  
12       28.   (Original) An obfuscation system as recited in claim 27, wherein  
13   the protection tools are selected from a group of protection tools comprising code  
14   integrity verification, acyclic code integrity verification, cyclic code integrity  
15   verification, secret key scattering, obfuscated function execution,  
16   encryption/decryption, probabilistic checking, Boolean check obfuscation, in-  
17   lining, reseeding pseudo random number generators with time varying inputs, anti-  
18   disassembly methods, varying execution paths between runs, anti-debugging  
19   methods, and time/space separation between tamper detection and response.

20  
21       29.   (Original) An obfuscation system as recited in claim 27, wherein  
22   the target segment selector comprises a pseudo random generator to enable  
23   random selection of the segment.  
24  
25

1           **30.**    (Original) An obfuscation system as recited in claim 27, wherein  
2 the tool selector comprises a pseudo random generator to enable random selection  
3 of the protection tool.

4  
5           **31.**    (Original) An obfuscation system as recited in claim 27, further  
6 comprising a quantitative unit to specify a quantity of protection qualities to be  
7 added to the digital good.

8  
9           **32.**    (Currently Amended) A client-server system, comprising:  
10 a production server to randomly apply various forms of protection only to  
11 selected portions of a digital good to produce a protected digital good; and  
12 a client to store and execute the protected digital good, the client being  
13 configured to evaluate the protected digital good to determine whether the  
14 protected digital good has been tampered with.

15  
16           **33.**    (Currently Amended) One or more computer-readable media  
17 having computer-executable instructions that, when executed, direct a computing  
18 device to:

19        parse a digital good into a plurality of segments; and  
20        apply multiple different protection tools to only a selected portion of the  
21 segments in a random manner to produce a protected digital good having a  
22 composite of variously protected portions.



1           34.    (Original) One or more computer-readable media as recited in claim  
2 33, further comprising computer-executable instructions to randomly select the  
3 protection tools from a set of available protection tools.

4  
5           35.    (Original) One or more computer-readable media as recited in claim  
6 33, further comprising computer-executable instructions to apply the protection  
7 tools to randomly selected portions of the original digital good.

8  
9           36.    (Original) One or more computer-readable media as recited in claim  
10 33, wherein the protection tools are selected from a group of protection tools  
11 comprising code integrity verification, acyclic code integrity verification, cyclic  
12 code integrity verification, secret key scattering, obfuscated function execution,  
13 encryption/decryption, probabilistic checking, Boolean check obfuscation, in-  
14 lining, reseeding pseudo random number generators with time varying inputs, anti-  
15 disassembly methods, varying execution paths between runs, anti-debugging  
16 methods, and time/space separation between tamper detection and response.